Лабораторна робота №3

Створення утіліти «DiskInfo»

Список усіх логічних дисків в системі.

Лістинг:

```
#include <iostream>
#include <Windows.h>

int main() {
    DWORD drives = GetLogicalDrives();

    for (char letter = 'A'; letter <= 'Z'; ++letter) {
        if (drives & 1) {
            std::cout << letter << ":\\" << std::endl;
        }
        drives >>= 1;
    }

    return 0;
}
```

Отримати тип кожного диску присутнього в системі, та дати пояснення для кожного типу диску.

Лістинг:

```
#include <iostream>
#include <Windows.h>

int main() {
    DWORD drives = GetLogicalDrives();

    for (char letter = 'A'; letter <= 'Z'; ++letter) {
        if (drives & 1) {
            std::string drive = std::string(1, letter) + ":\\";
            DWORD driveType = GetDriveType(drive.c_str());

            std::string typeString;
            switch (driveType) {
                case DRIVE_UNKNOWN:
                    typeString = "Не відомий тип";
                    break;
                case DRIVE_NO_ROOT_DIR:
                    typeString = "Диск без кореневого каталогу";
                    break;
                case DRIVE_REMOVABLE:
                    typeString = "Видалний диск";
                    break;
                case DRIVE_FIXED:
                    typeString = "Фіксований диск";
                    break;
                case DRIVE_REMOTE:
                    typeString = "Мережевий диск";
                    break;
                case DRIVE_CDROM:
                    typeString = "CD-ROM";
                    break;
                case DRIVE_RAMDISK:
                    typeString = "RAM диск";
                    break;
                default:
```

```
                    typeString = "Невідомий";
            }

            std::cout << letter << ":\\ - " << typeString << std::endl;
        }
        drives >>= 1;
    }

    return 0;
}
```

У цьому прикладі використовується функція GetDriveType, яка повертає значення типу диску. Потім за допомогою switch-case конструкції визначається тип диску і виводиться відповідне пояснення.

Для отримання інформації про диски в системі та файлові системи, які використовуються на них, можна скористатися функцією GetVolumeInformation, яка дозволяє отримати інформацію про том, на якому знаходиться вказаний каталог.

Лістинг:

```
#include <iostream>
#include <Windows.h>
#include <string>

int main() {
    DWORD drives = GetLogicalDrives();

    for (char letter = 'A'; letter <= 'Z'; ++letter) {
        if (drives & 1) {
            std::string drive = std::string(1, letter) + ":\\";
            char volumeName[MAX_PATH + 1] = {0};
            char fileSystemName[MAX_PATH + 1] = {0};
            DWORD serialNumber = 0;
            DWORD maxComponentLength = 0;
            DWORD fileSystemFlags = 0;

            if (GetVolumeInformationA(drive.c_str(), volumeName,
sizeof(volumeName), &serialNumber, &maxComponentLength, &fileSystemFlags,
fileSystemName, sizeof(fileSystemName))) {
                std::cout << "Диск " << letter << ":\\ - ";
                std::cout << "Назва тома: " << volumeName << ", ";
                std::cout << "Серійний номер: " << serialNumber << ", ";
                std::cout << "Файлова система: " << fileSystemName << std::endl;
            }
        }
        drives >>= 1;
    }

    return 0;
}
```

Тут використовується функція GetVolumeInformationA, яка повертає інформацію про файлову систему та інші параметри тома. Потім ця інформація виводиться на екран.

Отримати інформацію про зайнятості та вільне місце на кожному з дисків.

Для отримання інформації про зайнятість та вільне місце на кожному з дисків в системі можна скористатися функцією GetDiskFreeSpaceEx. Ця функція повертає інформацію про доступне і зайняте простори на диску.

Лістинг:

```
#include <iostream>
#include <Windows.h>
#include <string>

int main() {
    DWORD drives = GetLogicalDrives();

    for (char letter = 'A'; letter <= 'Z'; ++letter) {
        if (drives & 1) {
            std::string drive = std::string(1, letter) + ":\\";
            ULARGE_INTEGER freeBytesAvailable;
            ULARGE_INTEGER totalNumberOfBytes;
            ULARGE_INTEGER totalNumberOfFreeBytes;

            if (GetDiskFreeSpaceExA(drive.c_str(), &freeBytesAvailable,
&totalNumberOfBytes, &totalNumberOfFreeBytes)) {
                std::cout << "Диск " << letter << ":\\ - ";
                std::cout << "Вільно: " << freeBytesAvailable.QuadPart << "
байт, ";
                std::cout << "Загалом: " << totalNumberOfBytes.QuadPart << "
байт" << std::endl;
            }
        }
        drives >>= 1;
    }

    return 0;
}
```

Отримати інформацію про системну пам'ять.

Для отримання інформації про системну пам'ять можна скористатися функцією GlobalMemoryStatusEx. Ця функція надає інформацію про використання фізичної та віртуальної пам'яті системи.

Лістинг:

```
#include <iostream>
#include <Windows.h>

int main() {
    MEMORYSTATUSEX memoryStatus;
    memoryStatus.dwLength = sizeof(memoryStatus);

    if (GlobalMemoryStatusEx(&memoryStatus)) {
        std::cout << "Фізична пам'ять:" << std::endl;
        std::cout << "Загальний обсяг: " << memoryStatus.ullTotalPhys << " байт"
<< std::endl;
        std::cout << "Вільно: " << memoryStatus.ullAvailPhys << " байт" <<
std::endl;
        std::cout << "Використано: " << memoryStatus.ullTotalPhys -
memoryStatus.ullAvailPhys << " байт" << std::endl;

        std::cout << std::endl;

        std::cout << "Віртуальна пам'ять:" << std::endl;
```

```
        std::cout << "Загальний обсяг: " << memoryStatus.ullTotalVirtual << "
байт" << std::endl;
        std::cout << "Вільно: " << memoryStatus.ullAvailVirtual << " байт" <<
std::endl;
        std::cout << "Використано: " << memoryStatus.ullTotalVirtual -
memoryStatus.ullAvailVirtual << " байт" << std::endl;
    } else {
        std::cerr << "Не вдалося отримати інформацію про системну пам'ять." <<
std::endl;
    }

    return 0;
}
```

Отримати інформацію про Назву комп'ютера

Лістинг:

```
#include <iostream>
#include <Windows.h>

int main() {
    TCHAR computerName[MAX_COMPUTERNAME_LENGTH + 1];
    DWORD size = sizeof(computerName) / sizeof(computerName[0]);

    if (GetComputerName(computerName, &size)) {
        std::wcout << L"Назва комп'ютера: " << computerName << std::endl;
    } else {
        std::cerr << "Не вдалося отримати назву комп'ютера." << std::endl;
    }

    return 0;
}
```

У цьому прикладі використовується функція GetComputerName, яка заповнює масив computerName назвою комп'ютера. Потім ця назва виводиться на екран.

Отримати Назву поточного користувача

Лістинг:

```
#include <iostream>
#include <Windows.h>
#include <Lmcons.h>

int main() {
    WCHAR userName[UNLEN + 1]; // Use WCHAR for wide characters
    DWORD size = sizeof(userName) / sizeof(userName[0]);

    if (GetUserNameW(userName, &size)) { // Use GetUserNameW for wide characters
        std::cout << L"Поточний користувач: " << userName << std::endl;
    }
    else {
        std::cerr << "Не вдалося отримати назву поточного користувача." << std::endl;
    }

    return 0;
}
```

Отримати інформацію про поточний системний каталог, Тимчасовий каталог, поточний робочий каталог.

Для отримання інформації про поточний системний каталог, тимчасовий каталог та поточний робочий каталог використовуйте наступні функції:

1. **Поточний системний каталог**: Використовуйте функцію `GetSystemDirectory`, яка повертає шлях до системного каталогу.
2. **Тимчасовий каталог**: Використовуйте функцію `GetTempPath`, яка повертає шлях до тимчасового каталогу.
3. **Поточний робочий каталог**: Використовуйте функцію `GetCurrentDirectory`, яка повертає шлях до поточного робочого каталогу.

Лістинг:

```
#include <iostream>
#include <Windows.h>

int main() {
    TCHAR systemDirectory[MAX_PATH];
    if (GetSystemDirectory(systemDirectory, MAX_PATH) > 0) {
        std::wcout << L"Поточний системний каталог: " << systemDirectory <<
std::endl;
    }

    TCHAR tempPath[MAX_PATH];
    if (GetTempPath(MAX_PATH, tempPath) > 0) {
        std::wcout << L"Тимчасовий каталог: " << tempPath << std::endl;
    }

    TCHAR currentDirectory[MAX_PATH];
    if (GetCurrentDirectory(MAX_PATH, currentDirectory) > 0) {
        std::wcout << L"Поточний робочий каталог: " << currentDirectory <<
std::endl;
    }

    return 0;
}
```